Sumário

[Comandos de Saída 4](#_Toc40973890)

[Variáveis: 4](#_Toc40973891)

[Regras: 4](#_Toc40973892)

[Tipos Primitivos 4](#_Toc40973893)

[Declaração: 4](#_Toc40973894)

[Comandos de entrada 5](#_Toc40973895)

[Operadores aritméticos 5](#_Toc40973896)

[Ordem de precedência 5](#_Toc40973897)

[Funções Aritméticas 6](#_Toc40973898)

[Operadores Relacionais 6](#_Toc40973899)

[Operadores Lógicos 6](#_Toc40973900)

[Ordem de Precedência Geral 7](#_Toc40973901)

[Estruturas condicionais 7](#_Toc40973902)

[Se (Condicional Simples) 7](#_Toc40973903)

[Se Senao 7](#_Toc40973904)

[Se (Condicional Alinhada) 8](#_Toc40973905)

[Estrutura Escolha 8](#_Toc40973906)

[Estruturas de Repetição 9](#_Toc40973907)

[Enquanto(While) 9](#_Toc40973908)

[Repita (DO WHILE) 9](#_Toc40973909)

[Para (For) 9](#_Toc40973910)

[Rotinas 11](#_Toc40973911)

Algoritmos são conjuntos de passos finitos e organizados que, quando executados, resolvem um determinado problema.

Proporção Aurea

Algoritmos Computacionais:

“São passos a serem seguidos por um módulo processador e seus respectivos usuários que, quando executados na ordem correta, conseguem realizar determinada tarefa.

Sistemas de Gestão

Internet Banking

Redes Sociais

Vídeos

Games

Lógica de programação - > Linguagem de programação -> Sistema completo

Ferramentas de Lógica –

Fluxograma , Nassi Shneiderman, Portugol

São formas de representar a logica por trás do programa que será criado posteriormente.

# Comandos de Saída

Escreva(“.............”)

Escreval(“...............”)

Escreva(“mensagem”, msg)

# Variáveis:

Alocação de espaços vazios de memória.

## Regras:

Devem começas com uma letra.

Não se pode usar qualquer símbolo, a não ser o \_.

Não podem conter espaços em branco.

Sem acentos.

Não pode ser uma palavra reservada.

## Tipos Primitivos

Inteiro Ex: 1 3 -5 198

Real Ex: 5.5 1.3 0.6

Caractere Ex: “Gustavo” “Algoritmo” “123”

Lógico Ex: Verdadeiro e falso

## Declaração:

Identificador: tipo

# Comandos de entrada

Leia (Variavel)

# Operadores aritméticos

|  |  |
| --- | --- |
| + | adição |
| - | subtração |
| \* | multiplicação |
| / | divisão |
| \ | divisão inteira |
| ^ | Exponenciação |
| % | Módulo (Resto da divisão) |

## Ordem de precedência

|  |  |
| --- | --- |
| () | parênteses |
| ^ | Exponenciação |
| \*/ | Multiplicação e Divisão |
| + - | Adição e Subtração |

## Funções Aritméticas

|  |  |  |  |
| --- | --- | --- | --- |
| Abs | Valor absoluto | Abs(-10) | 10 |
| Exp | Exponenciação | Exp(3,2) | 9 |
| Int | Valor inteiro | Int(3.9) | 3 |
| RaizQ | Raiz Quadrada | RaizQ(25) | 5 |
| Pi | Retorna Pi | Pi | 3.14 |
| Sen | Seno (rad) | Sen(0.523) | 0.5 |
| Cos | Cosseno (rad) | Cos(0.523) | 0.86 |
| Tan | Tangente (rad) | Tan(0.523) | 0.57 |
| GraupRad | Graus para Rad | GraupRad(30) | 0.523 |

# Operadores Relacionais

|  |  |
| --- | --- |
| > | Maior que |
| < | Menor que |
| >= | Maior ou igual a |
| <= | Menor ou igual a |
| = | Igual |
| < > | Diferente de |

# Operadores Lógicos

E

OU

Não

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** | **A e B** | **A ou B** |  |  |
| V | V | V | V |  |  |
| V | F | F | V |  |  |
| F | V | F | V |  |  |
| F | F | F | F |  |  |

# Ordem de Precedência Geral

|  |  |
| --- | --- |
| Aritméticos | () |
| ^ |
| \*/ |
| +/- |
| Relacionais | Todos |
| Lógicos | E |
| Ou |
| Não |

# Estruturas condicionais

## Se (Condicional Simples)

Se (Expressão) então

Bloco

FimSe

## Se Senao

Se (Expressão) então

Bloco

Senao

Bloco

FimSe

## Se (Condicional Alinhada)

Se (Expressão) então

Bloco

Senao

Se (Expressão) então

Bloco

Senao

Bloco

....

FimSe

## Estrutura Escolha

Escolha ( variável)

Caso valor

Bloco A

Caso valor

Bloco B

Caso valor

Bloco C

OutroCaso

Bloco D

Fim Escolha

OutroCaso é opcional

Um caso pode ter mais de uma escolha:

Escolha ( variável)

Caso 1 , 2 , 3

Bloco A

# Estruturas de Repetição

## Enquanto(While)

Sintaxe: Enquanto expressão faça

Comando

FimEnquanto

Exemplo:

Mao<- 0

Enquanto(mao<=5) faça

Troca

Mao<-mao+1

FimEnquanto

## Repita (DO WHILE)

Sintaxe :

Repita

Bloco

Ate condição

## Para (For)

Sintaxe :

Para variável <- inicio ate fim[passo salto] faca

Bloco

FimPara

Exemplo :

Para C <- 1 ate 10 passo 1 faca

EscrevaL(C)

FimPara

# Rotinas

## Procedimento

Ex:

Procedimento Nome ()

Bloco

FimProcedimento

## Passagem de parâmetro por VALOR

Procedimento Soma(A,B:Inteiro)

Inicio

Bloco

FimProcedimento

Ex:

Algoritmo: C:\Users\brusa\Desktop\Curso Algoritmos e Lógica\03 - ParImparProcedimento.alg

Data: 21/05/2020

1 algoritmo "ParOuImparProcedimento"

2 // =============================================================

3 // Algoritmo para verificar se um valor e par ou impar usando

4 // rotinas internas.

5 //

6 // AUTOR: Gustavo Guanabara

7 // Curso COMPLETO disponivel GRATIS no site www.cursoemvideo.com

8 // =============================================================

9 var

10 N: Inteiro

11 Procedimento ParOuImpar(V: Inteiro)

12 inicio

13 Se (V%2 = 0) entao

14 EscrevaL("O numero ", V, " e PAR")

15 senao

16 EscrevaL("O numero ", V, " e IMPAR")

17 FimSe

18 FimProcedimento

19 inicio

20 Escreva("Digite um numero: ")

21 Leia (N)

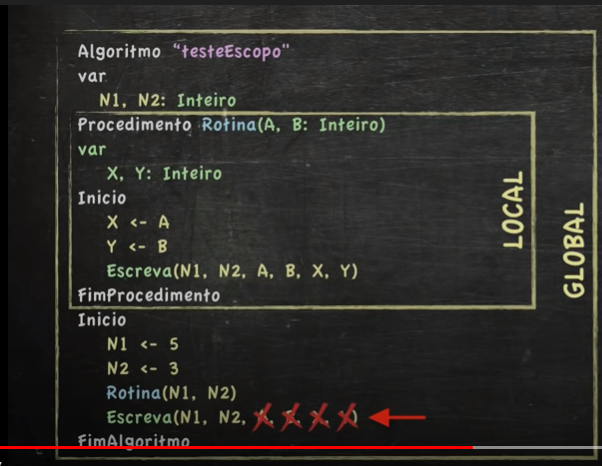
22 ParOuImpar(N)

23 fimalgoritmo

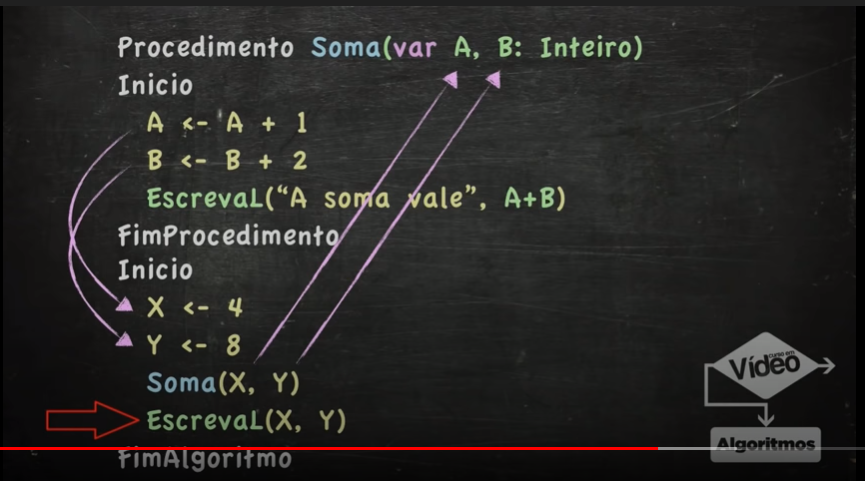
# Escopo

Variáveis declaradas dentro de procedimentos, só funcionam lá.

Variáveis globais podem ser chamadas mesmo quando foram declaradas dentro do procedimento.



## Passagem de parâmetro por REFERÊNCIA



Nesse caso, o procedimento se apropria das variáveis X e Y para fazer o calculo do procedimento. No Parâmetro por VALOR, o procedimento apenas atribui o valor das variáveis X e Y a A e B, mas elas continuam sendo coisas diferentes. No parâmetro por referência, por o procedimento se apropriar das variáveis, elas assumem o novo valor que foi processado no procedimento.